Multimed Tools Appl (2010) 50:29–47
DOI 10.1007/s11042-009-0365-x

Multi-object tracking evaluated on sparse events

Daniel Roth · Esther Koller-Meier · Luc Van Gool

Published online: 26 September 2009

Abstract This article presents a visual object tracking method and applies an event-based performance evaluation metric for assessment. The proposed monocular object tracker is able to detect and track multiple object classes in non-controlled environments. The tracking framework uses Bayesian per-pixel classification to segment an image into foreground and background objects, based on observations of object appearances and motions in real-time. Furthermore, a performance evaluation method is presented and applied to different state-of-the-art trackers based on successful detections of semantically high level events. These events are extracted automatically from the different trackers an their varying types of low level tracking results. Then, a general new event metric is used to compare our tracking method with the other tracking methods against ground truth of multiple public datasets.

Keywords Tracking · Event detection · Performance evaluation · Real-time

1 Introduction

Tracking moving objects in video data is part of a broad domain of computer vision that has received a great deal of attention from researchers over the last twenty years. This gave rise to a body of literature, of which surveys can be found in the work of Moeslund et al. [12, 13], Valera and Velastin [21] and Hu et al. [9]. Computer-vision-based tracking has established its place in many real world applications; among

D. Roth (✉) · E. Koller-Meier · L. Van Gool
Computer Vision Laboratory, ETH Zurich,
Sternwartstrasse 7, CH-8092 Zurich, Switzerland
e-mail: droth@vision.ee.ethz.ch

E. Koller-Meier
e-mail: ebmeier@vision.ee.ethz.ch

L. Van Gool
e-mail: vangool@vision.ee.ethz.ch

these are: visual surveillance, analysis of sports, video editing, tracking of laboratory animals, human-computer interfaces and cognitive systems.

In the first part of this article, the aim is to develop an intelligent visual object tracking method for real-time surveillance in a single camera. The general challenges of a multi-object tracker are: object detection, discriminative appearance modeling, tracking the different objects and handling their occlusions. In our approach we address each of these challenges via separated and specialized modules. Particularly, the monocular view and the real-time constraints make the problem challenging. Our contributions are the special modifications to each module for the integration into a Bayesian per-pixel classification framework in order to find a good balance between robustness and speed.

The real-time visual tracking method presented improves our previous techniques [16] by the following main contributions:

- New objects are detected by accumulating evidence on the calibrated ground plane by mapping foreground image regions to the real-world positions.
- Different object classes can be recognized by their distinctive footprint on the ground plane.
- Object segmentation is improved by means of an iterative object placement process. Knowledge about already found objects closer to the camera is used to refine prior probabilities of objects further away.

In the second part of this article we do not only compare and evaluate our tracker against others, but also present novel measurements and tools for this comparison. Performance evaluation of multi-object trackers for surveillance is itself a difficult problem and it has received significant attention in the form of tracking evaluation programs and workshops. Among the most important is the *Performance Evaluation of Tracking and Surveillance* (PETS) program [23] which started in 2000 with its first workshop as well as other programmes such as *Computers in the Human Interaction Loop* (CHIL) [4] or the ETISEO project [7].

As can be seen, evaluation programmes and metrics for video surveillance are almost as numerous as multi object tracking methods themselves. A problem is that they mostly address issues which do not directly tie in with the overall semantic interpretation of the scene that users would be mostly interested in. As an example, assessments of pixel-precise target detection are relevant for the evaluation of sub-components like figure-ground segmentation, but fall short of determining whether a system can make sense of what is going on in the scene.

Our novel tracking performance evaluation method uses sparse events in order to evaluate tracking performance on a higher conceptual level. Events such as *entering the scene*, *occlusion* or *picking-up a bag* are automatically extracted from the available tracking results. The metric then focuses on the completeness of such event detection to do the evaluation.

The rest of the article is organized as follows: First we state the most closely related work in multi-object tracking. Then, in Section 2 we introduce our tracking method and its probabilistic framework. The event extraction and metric is described briefly in Section 3. Section 4 shows the experimental results, and Section 5 discusses our findings and draws a conclusion.

1.1 Previous work

Majority of trackers are not based on segmentation, so they are applicable even for a moving camera. However, the absence of a segmentation typically requires a manual initialization of the tracked target, such as in the tracker by Comaniciu et al. [5] and Nummiaro et al. [14]. Okuma et al. [15] overcome this drawback by using a trained object detector for the initialization. Recent literature shows growing attention towards more complex detection based tracking approaches [11, 22], where trained object detectors identify target objects in every frame for further temporal analysis and tracking. However, when it comes to real-time application, trained detectors for humans or cars are still too slow and therefore not suitable for online real-time applications.

More closely related to our work are the following two publications of real-time trackers. Zhao et al. [24] present a real-time multi-camera tracker able to robustly down-project foreground pixels onto the assumed ground plane thanks to the use of stereo cameras. In contrast, our method only uses a monocular view of the scene. Lanz et al. [10] have developed a hybrid joint-separable formulation to model the joint state space of a multi-object tracker. While efficient and robust especially during occlusion their histogram model needs a careful initialization from different views prior to tracking. Our method in contrast learns a less specific but sufficient appearance model from a single view only at any location in the image.

In Section 4 we compare our method against two state-of-the-art trackers from Rowe et al. [18] and Duizer and Hansen [6]. Similar to our method, both use a segmentation but differ in the number of required cameras, their computational complexity or a complex hierarchical object management. More details are discussed in the related Subsections 4.2.1 and 4.2.3.

2 Multi-object tracking

In this Section we describe our real-time multi-object tracking algorithm including the object models, new object detection and the image segmentation. The proposed method performs a per-pixel classification to assign every pixel to one of the different objects that have been identified, including a background. The classification is based on the probability that a given pixel belongs to one of the objects given its specific color and position. The object probabilities are determined on the basis of two components. On the one hand, the appearance of the different objects is learned and updated, and yields indications of how compatible observed pixel colors are with these models. On the other hand, a motion model makes predictions of where to expect the different objects, based on their previous positions. The approach is akin to similar Bayesian filtering approaches, but has been extended by a novel object-class detection, an iterative segmentation refinement and a camera calibration.

Different characteristics for every object such as its appearance and motion are incorporated in specialized models and updated over time. Figure 1 sketches the tracking framework showing the appearance probability images for each object as well as an initial segmentation. Formally, the classification is described by Eqs. 1 and 2 below. The probability of a pixel to belong to the ith object o_t^i at time t is determined on the basis of an observation likelihood given the associated

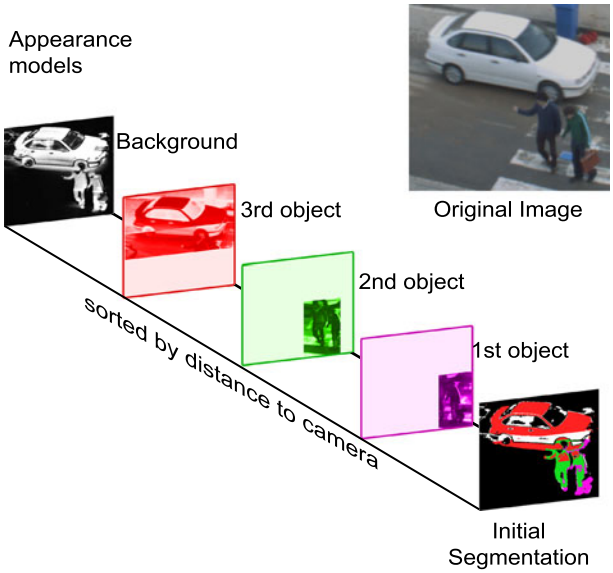


Fig. 1 Tracking framework: the maximum probability of the individual appearance models results in an initial segmentation using Bayesian per-pixel classification. The *white pixels* in the segmentation refer to the generic 'new object model' described in Section 2.5

object model. For all known objects including the background their probabilities to occupy a specific pixel location are calculated and compared. Using Bayes law, we can compute the posterior as a product of the appearance likelihood and a prior probability.

$$P_{posterior}\left(o_t^i|pixel_{1:t}\right) \propto P\left(pixel_t|o_t^i\right) P_{prior}\left(o_t^i|pixel_{1:t-1}\right) \tag{1}$$

$$segmentation = \max_{object}\left(P_{posterior}\left(o_t^i|pixel_{1:t}\right)\right) \tag{2}$$

- State vector for n objects : $objects = \{o^0, o^1, ..., o^n\}$
- Appearance model (likelihood): $P(pixel_t|o_t^i)$

In a first step the prior probability is the same for all objects. The segmentation assigns every pixel to the object with the maximum posterior probability. In the initial segmentation this only takes the appearance likelihoods given by our color models into account. The state vector contains the position and velocity on the ground floor of each foreground object.

The next Section introduces the tracking algorithm in more detail. It shows that the initial segmentation is further refined iteratively by modifying the prior probability of occluded objects.

2.1 Tracking algorithm

The tracker executes the steps shown in Table 1 for every frame. First, positions of known foreground objects are predicted and sorted according to their distance to the

Table 1 Tracking algorithm

1.	Predict and sort new object positions
2.	Compute appearance likelihood
3.	Initial segmentation of image by max probability
4.	Iterative object placement and segmentation image refining, from close to far objects
	4.1 Find object's new position in the image, by max window search
	4.2 Remove outlier pixels, boost inlier pixels and refine segmentation
5.	Detecting new objects and removing invisible ones
6.	Update all appearance models

camera. In a second step, the appearance models are applied to a small region—for computational reasons—around the predicted object position to compute the appearance probabilities for each object. An initial segmentation then assigns every pixel to the object with the highest appearance probability. The fourth step finds the new object positions (4.1) and refines the segmentation given the new object position (4.2). This is described in more detail in Section 2.4. The fifth step then searches for new objects, and deletes unseen ones before all appearance models are updated in the last step.

2.2 Tracking models

This Section briefly describes the different models used to compute the probabilities of the appearance models, as well as the dynamic model for the motion prediction.

Color model All our appearance models use variations of Gaussian mixtures in RGB color space. Stauffer and Grimson [19] have proposed this popular choice for modeling scene backgrounds with **t**ime-**a**daptive **p**er-**p**ixel **m**ixtures **o**f **G**aussians (TAPPMOGs). However, we have modified this approach to fit into our multi-model approach. The Gaussian is split among the background and foreground models described below.

Appearance background model In contrast to Stauffer's algorithm which combines foreground and background in one model we use one single Gaussian for each pixel of the background. The model is initialized at start-up with a clean background.

Appearance foreground model For the appearance models we use a 'sliced object model', as it divides the object into a fixed number of horizontal slices of equal height. For each slice, color models with multiple Gaussians are generated using EM representing the most important colors for that part of an object. Each object class such as pedestrians have a specific height and width of the object model.

Dynamic model The movement of each foreground object is predicted individually. We use a linear Kalman filter to model the movement on the ground plane in world coordinates.

2.3 Ground plane assumption

In addition to our previous method [16], we have added an extrinsic camera calibration [20]. In conjunction with a ground plane assumption, object movements are restricted and predicted onto the ground plane in world coordinates rather than 2D image coordinates. An example of the ground plane assumption is shown in Fig. 2. Furthermore, objects are assumed to have a fixed 3D size. We approximate the width and height of a human with a fixed sized cylinder resulting in hard constrains for the size of the bounding box. While this is a simplification, the system is still able to handle varying human shapes. The fixed object size in combination with the restriction to ground plane movements improved significantly the tracking as well as the localisation of the objects in world coordinates, especially under occlusion.

2.4 Iterative object placement and segmentation refinement

Our novel approach for finding the new object positions is based on an initial segmentation according to the maximum posterior probability (Eq. 2). It places objects iteratively from close to distant objects on the ground plane in world coordinates. Figure 3 visualizes this process. For each object the following two processing steps are applied where first the object's position is found and secondly, the segmentation is refined.

The new object position is found by applying a maximum window search on the 2D segmentation image. The bounding box width and height at the predicted object location is used to search over the whole segmentation image. The location of the maximum window search is taken as the new object position.

During the refining step some pixels of the segmentation image are modified. In Eq. 1 we lower the prior probabilities of pixels outside of the new bounding box, while increasing it inside. Equation 2 is then re-evaluated given the changed probabilities resulting in a refined segmentation. Pixels outside and initially labeled as the object are assigned to the object with the next highest probability. Furthermore, pixels inside of the bounding box of the new object position which are not

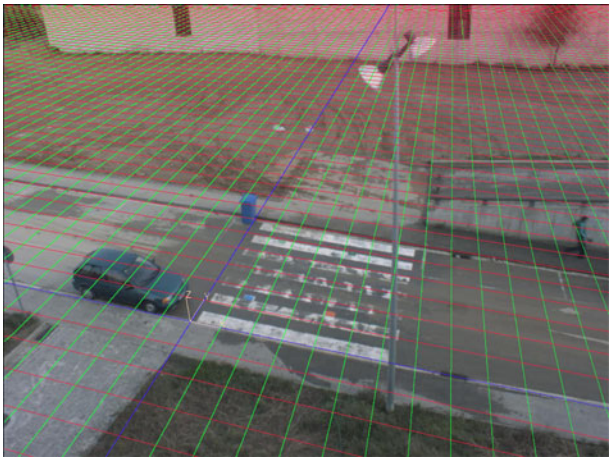


Fig. 2 Ground plane calibration

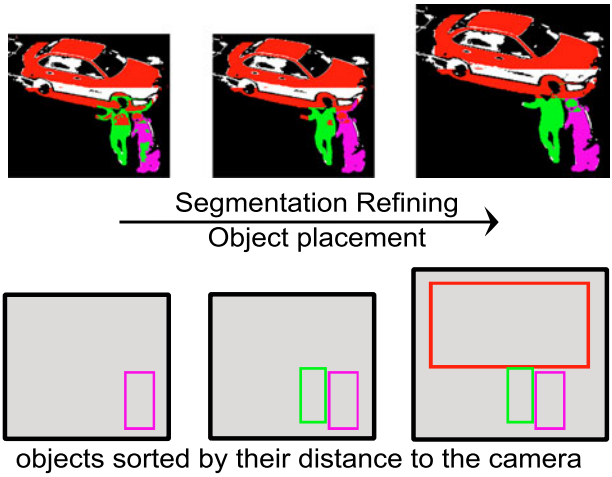


Fig. 3 While iteratively searching for the exact object position from close to distant object the segmentation is refined

yet segmented as the object, obtain a higher probability. The overall quality of the segmentation is improved, especially during occlusions.

2.5 New object detection

The tracker creates new foreground objects based on a generic 'new object model' $\mathcal{N}$. This special appearance model has a uniform, low probability $p_{\mathcal{N}}$. Thus, when the posterior probabilities of the background and all known foreground objects drop below $p_{\mathcal{N}}$, the pixel is assigned to $\mathcal{N}$ indicating a new object or a badly modeled foreground object.

By exploiting the camera calibration only pixels on $\mathcal{N}$ are projected onto the ground plane to vote for a new object position. Votes are generated as follows:

- One vote for every pixel
- Pixel votes are summed in a column of successive pixels of $\mathcal{N}$ giving the highest weight to the bottom pixel

With this voting scheme we accumulate evidence about new object positions on the ground plane. In conjunction with the camera calibration it allows us to map 2D image blobs to 3D world coordinates. Furthermore, it makes it possible to guess the depth of objects from a monocular camera, given a rough segmentation. Limitations are encountered for crowds, where the separation of individual objects on the ground plane might fail.

When the votes on the ground plane are determined a maximum window search is performed on the ground plane with the size of the expected object classes. As shown in Section 4 the method was successfully tested to distinguish between people and cars for different dataset. In this scenario, first a search with the maximum window size for the larger car is performed. Afterwards, we search for pedestrians with a smaller window on the remaining ground plane votes. The cumulated ground plane votes for a given object position and window size form a score which is compared to the initialization threshold of the object class. Furthermore, the new object position is checked for a possible overlap with current objects, which would prevent the

initialization of a new object at the same position. Additional boundary conditions prevent objects from being initialized at image borders, where correct object class distinction might not be possible.

The new object's position on the ground plane and object class directly determine the size of the bounding box. We assume a fixed real world height and width off all objects of the same class. All $\mathcal{N}$ pixels inside the box are removed from the segmentation image and their votes are removed from the ground plane. These pixels are then used to initialize the appearance model of the new object.

The maximum window search on the ground plane is repeated until no more new objects can be found of a certain class. Then the search for the next smaller object class starts.

3 Event-based performance evaluation

In this second part of the article we present our novel performance evaluation method [17] based on events, which we apply in the results Section. Figure 4 gives an overview of the event-based evaluation.

The evaluation metric is based on comparing the list of events extracted from ground truth data and the list of events extracted from the trajectories generated by the tracking algorithm. Therefore, the evaluation with these higher-level events directly targets the overall semantic interpretation of the scene. Evaluation methods on the pixel-level [1], frame-level [1, 2] or object trajectory level [2, 22] in contrary would only target the evaluation of sub-components of an algorithm.

Our event metric comprises the following additional advantages:

- The lengths of trajectories do not influence the metric making it independent of the frame rate and density of the ground truth labeling.
- It enables the fast generation of ground-truth data as not every frame needs to be annotated in full detail, as long as the events can be reliably extracted from sparse annotation.
- Reuse of already available ground truth data by automatic conversion into our novel event-based representation.
- Minimizing the human factor within the ground truth data and its influence onto the metric by means of event-based evaluation on a higher level.

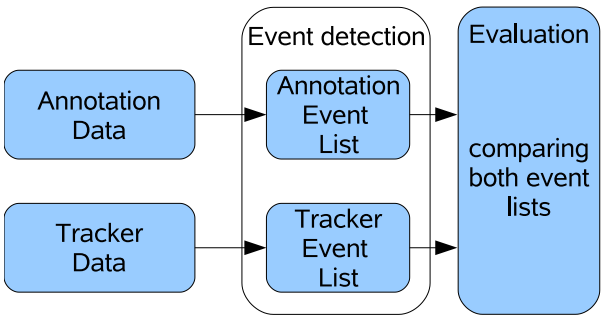


Fig. 4 Evaluation scheme

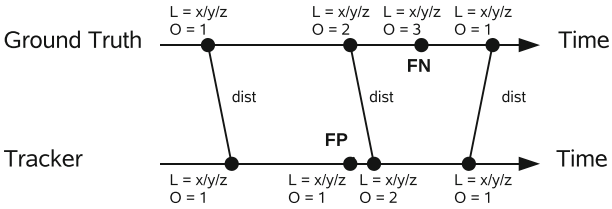


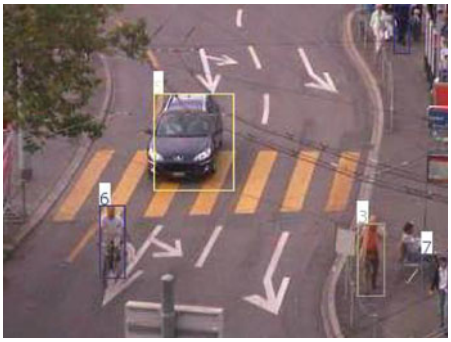
Fig. 5 Event matching between the same type by dynamic programming

- Establishing a least common denominator to represent tracking data which is versatile to handle many different output formats.
- The metric directly helps to improve tracking algorithms by identifying:
  - difficult trajectories
  - difficult scene locations
  - difficult situations
  - difficult event types
- Easy integration into higher level event and object detection frameworks.

An 'event' always describes instant actions happening at one particular point in time. In this work we automatically extract the following types of events from the tracking results: *entering the scene*, *leaving the scene* and *entering a pedestrian crossing*. All these events are triggered by exploiting the object positions in the tracking results. An event $\boldsymbol{E}$ is a 4-tuple and consists of an event-type $\mathcal{P}$, a point in time $\mathcal{T}$, a location $\mathcal{L}$ and it is related to one object $\mathcal{O}$. In order to automatically generate events from either manually labeled data or continuous tracker output, these four basic building blocks have to be identified. For each tracker we use individual conversion methods to generate the events depending on the type and format of the underlying data. As an example, *entering the scene* events are found in the tracking results where targets with new IDs appear the first time.

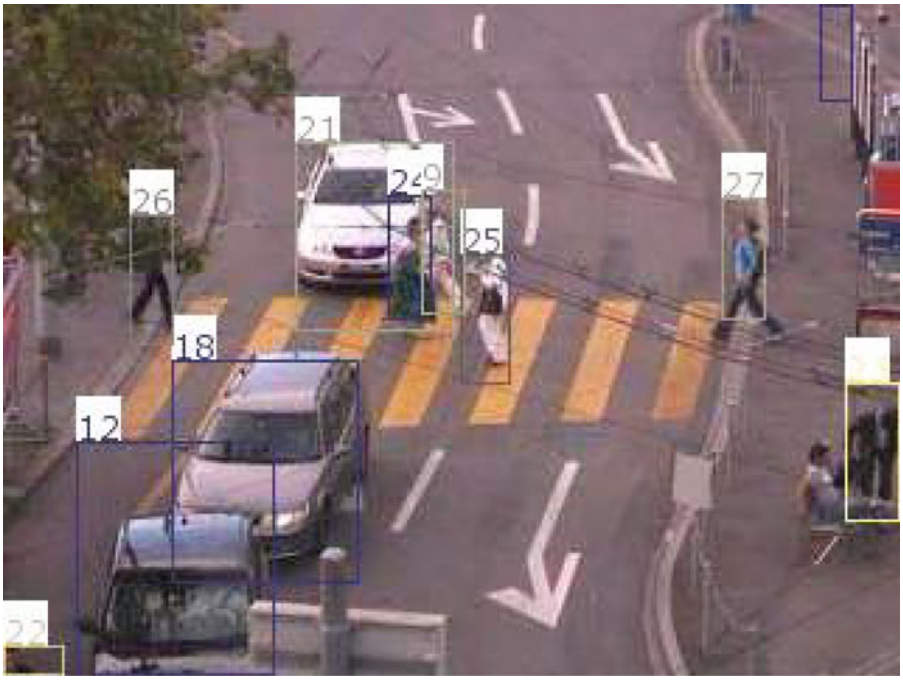
(a) (b)

Fig. 6 Segmentation from the Central square sequence. The different size of the bounding box visualizes the different object classes. Unique object IDs are shown in the top left corner. *Black pixels* = background, *white pixels* = unassigned pixels $\mathcal{N}$, *colored pixels* = individual objects. **a** Camera view. **b** Segmentation

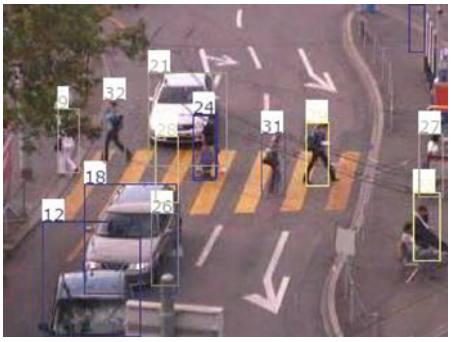


(a)

(b)



(c)



(d)

(e)

◀ **Fig. 7** Tracking results from the Central square sequence. Objects are visualized by their bounding box and unique ID in the top left corner. **a** Frame 454: Cyclist is tracked as a person. **b** Frame 1426: increasing number of objects. **c** Frame 1466 severe occlusion. **d** Frame 1581 several ID changes among pedestrians occurred. **e** Frame 1795: two people (ID 36) are mistaken as a car

Events are extracted from the ground truth data and from the tracker, as shown in Fig. 4. The two event lists are then compared by using dynamic programming and the event metric proposed in [17]. Events are matched according to the time and location distance as shown in Fig. 5. Matching events from the two lists are counted as true positives (TP). Events on the ground truth list without a corresponding event in the tracker list are counted as a false negative (FN). Unmatched events from the tracker are counted as false positives (FP).

As the distance measurement between two events i, j we combine $\mathcal{T}$ and $\mathcal{L}$ into one distance as described in Eq. 3.

$$dist_{i,j} = min(\alpha|\mathcal{T}_i - \mathcal{T}_j| + ||\mathcal{L}_i - \mathcal{L}_j||, maxdist) \tag{3}$$

Where $||...||$ is the Euclidean distance, α is a scaling factor in order to allow to compare the different units of seconds and meters. The parameter *maxdist* is a maximal distance above which the match is considered to have failed.

For our experiments we have chosen a combined maximal distance measure of 5 seconds respectively 12 meters. Furthermore, we compute the average time $\mathcal{T}_{ave}$ and location $\mathcal{L}_{ave}$ deviations from ground truth of all correct matches (TP) to measure the accuracy of the trackers. Finally, in a object-based evaluation we measure how many of all events were detected correctly for an individual ground truth object. There, we also count the total number $\mathcal{O}_{tot}$ of different $\mathcal{O}$ for these objects, in order to measure identity changes.

4 Results

This Section presents results of the tracking method as well as the event extraction on multiple public datasets. First, results of the object classification and tracking are discussed on a challenging sequence with cars and pedestrians. Due to the complexity of the sequence we also briefly compare our results with detection-based tracking approaches. Finally, we compare our tracker against two other state-of-the-art trackers on the CVC outdoor dataset. For the evaluation and comparison we use the event-based evaluation metric.

Table 2 Detected events for the CVC outdoor sequence

Events	Ground truth	Tracker:		
		1	2	3
Entering scene	8	14	7	8
Leaving scene	7	10	6	7
Entering pedX	6	6	6	6
Leaving pedX	7	7	6	6

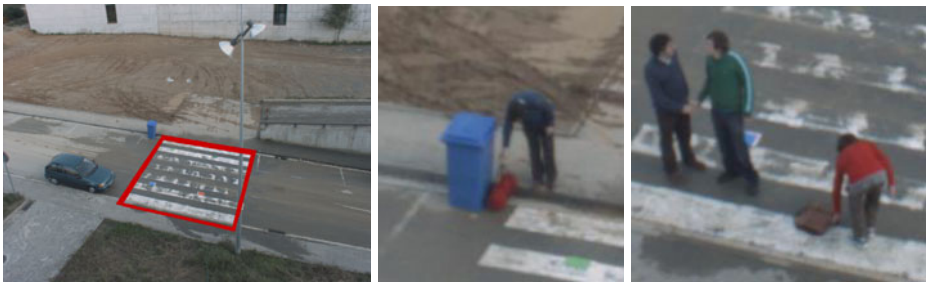


Fig. 8 CVC outdoor sequence. **a** Pedestrian crossing area used to trigger events. **b** Dropping 1st bag. **c** Picking-up 2nd bag

4.1 Central sequence

The Central pedestrian crossing sequence was recorded with a public web-cam at 15fps, 320 × 240 pixels resolution, and contains severe MPEG compression artifacts. Leibe et al. [11] previously presented results with a detector-based tracking on the same publicly available dataset [3]. Major challenges for the tracker are the correct classification between pedestrians and cars as up to a dozen objects lead to crowded situations and multiple occlusions.

Figure 7 shows the sequence with the results of the tracker overlaid. The refined segmentation is displayed in Fig. 6. During the first 800 frames all pedestrians and cars are correctly classified while entering the scene. Cyclists and bikers (Fig. 7a) are classified as pedestrians and all trajectories are correct. We equally divide the object height in five slices to learn the color models of the foreground objects. Despite specularities and reflections of cars the color models are able to constantly adapt to the changing appearance thanks to the EM learning.

In the following frames the number of cars and pedestrians constantly increases. Between frames 1400 in Fig. 7b and frame 1600 in Fig. 7d about 10 pedestrians cross the street from both sides while several cars are waiting. During this part of the sequence our tracker looses track of several of the pedestrians during the

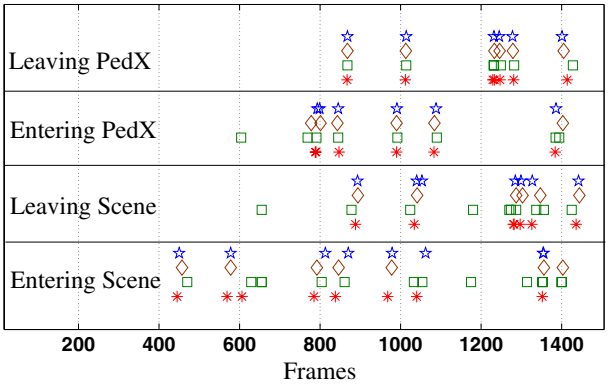


Fig. 9 Frames/Event plot for the CVC outdoor sequence. Stars equal ground truth, squares equal tracker1, diamonds equal tracker2 and pentagrams equal tracker3

Table 3 Event-based evaluation of the CVC outdoor sequence

	TP	FN	FP	$\mathcal{T}_{ave}$	$\mathcal{L}_{ave}$
Events tracker 1					
Entering scene	8	0	6	2.44s	2.30m
Leaving scene	7	0	3	0.67s	1.77m
Entering pedX	6	0	2	0.35s	0.30m
Leaving pedX	7	0	0	0.22s	0.43m
Events tracker 2					
Entering scene	6	2	1	0.56s	1.34m
Leaving Scene	6	1	0	0.59s	0.39m
Entering pedX	6	0	0	0.50s	0.80m
Leaving pedX	6	1	0	0.16s	0.64m
Events tracker 3					
Entering scene	7	1	1	1.04s	2.34m
Leaving scene	6	1	1	0.26s	0.54m
Entering pedX	6	0	0	0.27s	0.21m
Leaving pedX	6	1	0	0.24s	0.75m

severe occlusion in the center of the pedestrian crossing. However, due to the object initialization described in Section 2.5 missed objects are detected again after the occlusion phase.

In frame 1795 of Fig. 7e two people with a suitcase are mistakenly identified as a car due to their wider footprint on the ground plane.

In comparison, the non real-time pedestrian-detector approach [11] showed fewer but more complete tracks in crowded situations, while our method detects and tracks nearly all objects including cars in less crowded situations in real-time.

4.2 CVC outdoor sequence

We have also applied the tracking evaluation metric to the CVC outdoor sequence [8]. For this sequence tracking results from three trackers were compared for the following events: *entering scene*, *leaving scene*, *entering pedestrian crossing* and *leaving pedestrian crossing*. Table 2 shows the total number of detected events for each tracker as well as the ground truth. During the sequence two bags are carried by different persons which are labeled in the human annotated ground truth but only tracked by one of the three methods. The area considered as the pedestrian crossing is shown in Fig. 8a, while Fig. 9 shows the plotted events over time. The events were automatically extracted from the hand labeled ground truth as well as

Table 4 Object-based evaluation of tracker 1 (CVC outdoor sequence)

$\mathcal{O}$ Tracker 1	TP percentage	$\mathcal{O}_{tot}$
GT object 1	4/4	3
GT object 2	4/4	1
GT object 3	1/1	1
GT object 4	4/4	1
GT object 5	4/4	1
GT object 6	4/4	1
GT object 7	3/3	1
GT object 8	4/4	3
Total	28/28 (100%)	

Table 5 Object-based evaluation of tracker 2 (CVC outdoor sequence)

$\mathcal{O}$ Tracker 2	TP percentage	$\mathcal{O}_{tot}$
GT object 1	4/4	1
GT object 2	4/4	1
GT object 4	4/4	1
GT object 5	4/4	1
GT object 6	4/4	1
GT object 8	4/4	1
Total	24/24 (100%)	

from the tracker results in various data formats including CAVIAR xml and other proprietary formats.

4.2.1 Tracker 1

The first segmentation based tacker by Rowe et al. [18] uses a modular and hierarchically organized tracking system. A set of co-operating modules, which follow both bottom-up and top-down paradigms, are distributed through three abstraction levels of tracking. Each level is devoted to one of the main different tasks to be performed: target detection, low-level tracking (short-term blob tracker), and high-level tracking. The latter embeds switching mechanism among different operation modes, namely *motion-based tracking* and *appearance-based tracking*.

Table 3 shows the results of that tracker. In comparison to the other two methods it is the only tracker with zero FN. Only tracker 1 detects and tracks the bags in the scene (see Figs. 8b, 8c and 11). However, these good results come with a price of multiple FP and identity changes during pickup and dropping of the bags as shown in Table 4. The disappearing car at the end of the sequence leads to multiple wrong object appearances increasing the number of objects overly.

4.2.2 Tracker 2

Tracker 2 is our method presented in this article in Section 2. As shown in Table 5 it perfectly tracks all pedestrians and cars without identity changes resulting in

(a) (b)

Fig. 10 Presented tracking method on CVC outdoor sequence. **a** Object tracking. **b** Segmentation

Fig. 11 CVC outdoor sequence: tracker 1–3 from left to right. Only tracker 1 detects the small bags (no. 3 and no. 23)

complete tracks. Cars and pedestrians are correctly classified resulting in the larger fixed object size for cars in comparison to humans as shown in Fig. 10. However, the bags and their events are completely missed and not tracked resulting in several FN shown in Table 3 and Fig. 11. In addition, a wrong object appears at the position of the parked car when it drives away at the end of the sequence.

4.2.3 Tracker 3

The third tracker by Duizer and Hansen [6] is a multi-view tracking system based on the planar homography of the ground plane. The foreground segmentation is performed using the codebook method for each view separately. Given an appropriate training, the codebook segmentation allows robust operation in a 24/7 situation capable of adapting to severe illumination changes. The tracking of objects is performed in each view using bounding box overlap, and occlusion situations are resolved by probabilistic appearance models.

For this experiment two cameras were used. Their overlapping region however was restricted to the pedestrian crossing and some parts of the road, which requires single view tracking in several areas of the scene.

The results of this tracker are quite similar to tracker 2 as it tracks cars and pedestrians but also ignores the bags. The wrong object initialized at the empty space of the disappearing car shows the same segmentation problem as the other trackers. However, results could be different if the codebook for the background would have been trained without this car. Furthermore, this tracker looses track of one pedestrian when it passes behind a lamp pole resulting in an identity change shown in Table 6 (GT Object 6). The lamp pole is only seen in one of the two cameras.

Table 6 Object-based evaluation of tracker 3 (CVC outdoor sequence)

$\mathcal{O}$ Tracker 3	TP percentage	$\mathcal{O}_{tot}$
GT object 1	4/4	1
GT object 2	4/4	1
GT object 4	4/4	1
GT object 5	4/4	1
GT object 6	4/4	2
GT object 7	1/3	1
GT object 8	4/4	1
Total	25/27 (93%)	

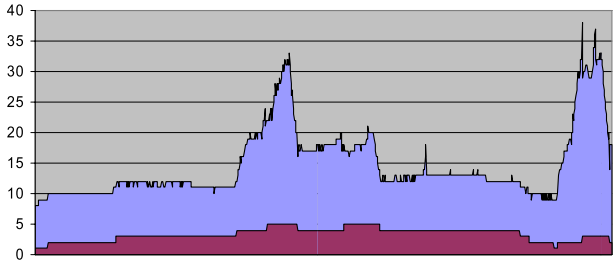


Fig. 12 Computational effort: the *blue curve* above shows the computation time in milliseconds per frame. Below in *red*, the number of tracked objects is given

In conclusion, the evaluation shows that each tracker has its own strengths and weaknesses. Only tracker 1 detects bags and therefore finds all events, but also some unwanted. Tracker 2 makes a perfect job tracking cars and pedestrians while not tracking any bags. Tracker 3 adds an identity change to the otherwise similar results in comparison to tracker 2. Disappearing objects initially learned as background cause problems with all three methods.

4.3 Performance

Figure 12 plots the computation time in milliseconds as well as the number of objects tracked for the CVC outdoor sequence of the proposed tracker. The time was measured on a 2.13 GHz CPU with a video resolution of 320×240. The time varies between 8 and 38 ms and scales with the number of foreground pixels in the scene. The two peaks directly indicate the presence of the larger cars while the number of pedestrians has a much lower impact on the computational cost. For a single frame, most time is spent for the computation of the pixel probabilities as well as the segmentation.

5 Conclusions

First, a novel tracking framework was introduced and tested on multiple challenging datasets in comparison with other state-of-the-art methods. Different object classes are recognized by their distinctive footprints by accumulating evidence on the calibrated ground plane. An iterative object placement process improves the segmentation and tracking, especially during inter-object occlusions. The capabilities and limitations of the real-time tracker were shown on multiple challenging datasets. Secondly, semantically high-level events such as "entering the pedestrian crossing" could be automatically extracted from the tracking results. Furthermore, the available annotated ground truth data from public datasets could be reused and converted automatically into our high-level representation. A comparison between different trackers and human annotated ground truth was carried out on the event level.

Acknowledgements The authors gratefully acknowledge support by the Swiss SNF NCCR project IM2 and EU project HERMES (FP6-027110). Furthermore, we would like to thank Prof. Dr. Thomas Moeslund from the University of Aalborg, Denmark and Dr. Jordi Gonzalez from the CVC Center in Barcelona, Spain for their tracking results and valuable input.

References

1. Aguilera J, Wildernauer H, Kampel M, Borg M, Thirde D, Ferryman J (2005) Evaluation of motion segmentation quality for aircraft activity surveillance. In: IEEE int workshop on VS-PETS, pp 293–300
2. Bashir F, Porikli F (2006) Performance evaluation of object detection and tracking systems. In: IEEE international workshop on PETS, vol 5, pp 7–14
3. Central pedestrian crossing: dataset. http://www.ee.ethz.ch/bleibe/data/datasets.html
4. Chil project website. http://chil.server.de/
5. Comaniciu D, Ramesh V, Meer P (2000) Real-time tracking of non-rigid objects using mean shift. In: CVPR
6. Duizer P, Hansen D (2007) Multi-view video surveillance of outdoor traffic (master thesis). In: Digital project library, Aalborg University, Denmark
7. Etiseo: video understanding evaluation. http://www.silogic.fr/etiseo
8. Gonzlez J, Roca FX, Villanueva JJ (2007) Hermes: a research project on human sequence evaluation. In: Computational vision and medical image processing (VipIMAGE'2007)
9. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. IEEE Trans Syst Man Cybern 34:334–352
10. Lanz O (2006) Approximate bayesian multibody tracking. IEEE Trans Pattern Anal Mach Intell 28(9):1436–1449
11. Leibe B, Schindler K, Van Gool L (2007) Coupled detection and trajectory estimation for multi-object tracking. In: International conference on computer vision (ICCV'07)
12. Moeslund TB, Granum E (2001) A survey of computer vision-based human motion capture. Comput Vis Image Underst 81(3):231–268
13. Moeslund TB, Hilton A, Kruger V (2006) A survey of advances in vision-based human motion capture and analysis. Comput Vis Image Underst 104:90–126
14. Nummiaro K, Koller-Meier E, Van Gool L (2003) An adaptive color-based particle filter. J Image Vis Comput 21(1):99–110
15. Okuma K, Taleghani A, Freitas Nd, Littlei JJ, Lowe DG (2004) A boosted particle filter: multitarget detection and tracking
16. Roth D, Doubek P, Van Gool L (2005) Bayesian pixel classification for human tracking. In: MOTION, pp 78–83
17. Roth D, Koller-Meier E, Rowe D, Moeslund T, Van Gool L (2008) Event-based tracking evaluation metric. In: IEEE workshop on motion and video computing (WMVC)
18. Rowe D, Reid I, Gonzlez J, Villanueva J (2006) Unconstrained multiple-people tracking. In: 28th DAGM. Springer LNCS, Berlin, pp 505–514
19. Stauffer C, Grimson W (1999) Adaptive background mixture models for real-time tracking. In: CVPR, pp 246–252
20. Tsai R (1987) A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. IEEE J Robot Autom 3(4):323–344
21. Valera M, Velastin S (2005) Intelligent distributed surveillance systems: a review. IEE Proc Vis Image Signal Process 152(2):192–204
22. Wu B, Nevatia R (2006) Tracking of multiple, partially occluded humans based on static body part detection. In: CVPR, pp 951–958
23. Young D, Ferryman J (2005) Pets metrics: on-line performance evaluation service. In: Proc. 2nd joint IEEE int workshop on VS-PETS, pp 15–16
24. Zhao T, Aggarwal M, Kumar R, Sawhney H (2005) Real-time wide area multi-camera stereo tracking. In: CVPR, pp 976–983

Daniel Roth received his MSc ETH in Electrical Engineering and Information Technology in 2004 from the Swiss Federal Institute of Technology (ETH Zurich). Currently, he is working as a Ph.D. candidate within the Computer Vision Group of the same university. His research interests include multi-object tracking, performance evaluation and real-time systems.

Dr. Esther Koller-Meier received her Master's degree in Computer Science in 1995 from the Swiss Federal Institute of Technology (ETH Zurich). At the beginning of 2000 she obtained her Ph.D. from the Department of Electrical Engineering of the same university. Currently, she is working as a Postdoc within the Computer Vision Group of the ETH Zurich. Her research interests include object tracking, gesture analysis and multi-camera systems.

Prof. Luc Van Gool is leader of the Computer Vision Lab at the ETH Zurich in Switzerland and the institute VISICS at the University of Leuven in Belgium. He has authored more than 200 papers in the field of computer vision. His main interests include 3D reconstruction and modelling, object recognition, grouping and segmentation, tracking and optical flow, robot navigation and registration. In 1998, he received the David Marr Prize and an EITC Prize from the European Commission. He has also received two TechArt awards and a Henry Ford prize for conservation of the environment. Author of several patents. Member of the editorial board of the ACM Journal on Computing and Cultural Heritage, the International Journal on Computer Vision, the IEEE Transactions on Pattern Analysis and Machine Intelligence, and Machine Vision and Applications. Editor-in-chief of the Journal Foundations & Trends in Computer Graphics and Computer Vision. Besides kooaba, cofounder of the companies eSaturnus (endoscopic imaging), Eyetronics (delivering 3D models to the CH sector), GeoAutomation (mobile mapping), and Procedural (3D modeling of buildings).